**Project 3**  **Due: June 13, 2017**

In this project, you will classify hand-writing digits by machine learning methods. In general, there are mainly two things you need to take care of in machine learning: First, extract features for data; second, train a classifier from features.

Since we focus on classifier in our class, in this project, you can use third-part library to extract features but you need to implement classifier by yourself. Two combinations of features and classifier will be implemented, i.e., binary image + naïve Bayes and LBP + SVM. **Matlab is recommended for this project.**

1. **Folder structure and dataset**

digit

-----data

-------loadMNISTImages.m

-------loadMNISTLabels.m

-------train-images-idx3-ubyte

-------train-labels-idx1-ubyte

-------t10k-images-idx3-ubyte

-------t10k-labels-idx1-ubyte

-----src

-------main\_naive\_bayes.m

-------main\_svm.m

-------lbp.m

The files start with train\* are train data and label, you can use loadMNISTImages.m and loadMNISTLabels.m to load them, similar, files start with t10k are test data. After you load images, you get a m\*n matrix, where each column is an image, you can reshape a column to 28\*28 to get a square image. The intensity of each pixel is rescaled to 0-1.

1. **Binary image + Naïve Bayes**

In the first combination of feature and classifier, we choose binary image to be the feature, i.e., you set nonzero intensities to be 1 and others remain to be 0. By doing so, you make the intensity to be discrete, which is crucial for naïve Bayes. Note the dimension of feature is the same as raw data because you only convert raw data to binary.

Then you implement Naïve Bayes classifier and use training data to train a model. After that, you can classify testing data and evaluate the accuracy.

1. **LBP + SVM**

LBP is a wildly used image feature. It can describe texture information in an image. In src folder, there is a lbp.m file to compute LBP feature for an image. Note, you need to reshape one data, i.e., one column of data matrix computed by loadMNISTImages.m to 28\*28 before you pass it to lbp.m.

Then you need to implement a SVM classifier and predict test data. The basic SVM you develop may be a binary class one, that is, only support two kinds of classes (positive and negative classes). However, for digit classification, you need to classify 10 classes, so you actually need a multi-class SVM. There are several ways to extend binary classifier to multi-class classifier, like “one vs all” and “one vs one”, see <https://en.wikipedia.org/wiki/Multiclass_classification> for more information.

1. **Submit your code**

The main callers should be main\_naive\_bayes.m and main\_svm.m, make sure these two files are runnable. If you need more third part libraries, add them in src folder and add an “addpath()” in main caller so that I can run your code without any modification or configuration.

Then zip src folder to **digit\_XXXXXXXX.zip** where XXXXXXXX stands for your student id.